FACULTY OF INFORMATION SCIENCE BACHELOR OF

SCIENCE (HONS) UNIVERSITI TEKNOLOGI MARA (UiTM)

SELANGOR BRANCH, PUNCAK PERDANA CAMPUS

ADVANCED WEB DESIGN DEVELOPMENT AND CONTENT MANAGEMENT (IMS566)

GROUP ASSIGNMENT:

WEB APPLICATION WITH DATABASE INTERACTION PROJECT

PREPARED BY:

1.	MEGAT NAUFAL SYABIL BIN ZAMRI	2025121211
2.	MUHAMMAD NAJMI BIN ISMADY	2025197825
3.	FARIS AFIZUAN BIN ABD KAHARMUZAKIR	2025136615
4.	MUHAMMAD AMIR AMSYAR BIN ZAIN AZMAN	2025197521

PREPARED FOR:

DR. MUHAMMAD ASYRAF BIN WAHI ANUAR

CLASS:

CDIM2624A

SUBMISSION DATE:

2nd FEBRUARY 2026

Advanced Web Design Development and Content Management

(IMS566)

Group Assignment Title

Flight Booking System (FlyHigh)

Prepared by

1.	MEGAT NAUFAL SYABIL BIN ZAMRI	2025121211
2.	MUHAMMAD NAJMI BIN ISMADY	2025197825
3.	FARIS AFIZUAN BIN ABD KAHARMUZAKIR	2025136615
4.	MUHAMMAD AMIR AMSYAR BIN ZAIN AZMAN	2025197521

FACULTY OF INFORMATION SCIENCE

BACHELOR OF SCIENCE (HONS)

UNIVERSITI TEKNOLOGI MARA (UiTM) SELANGOR

BRANCH, PUNCAK PERDANA CAMPUS

ACKNOWLEDGMENT

بسم الله الرحمن الرحيم

Alhamdulillah. All gratitude belongs to Allah SWT for granting us the ability to finish this report and for His willingness to do so. We can finish this report with a lot of energy and determination with His blessings.

Assalamualaikum w.b.t. We, Megat Naufal Syabil Bin Zamri, Muhammad Najmi Bin Ismady, Faris Afizuan Bin Abd Kaharmuzakir and Muhammad Amir Amsyar Bin Zain Azman, owe a debt of gratitude to the following people who have contributed to the success of our assignment. We would like to thank them for all their valuable help on our preparation by finishing our project system report.

In fact, we would like to say thank you to our lecturer, Dr. Muhammad Asyraf Bin Wahi Anuar as our lecturer of this subject, Advanced Web Design Development And Content Management (IMS566) who has guided us in doing this task since the day one and especially for answering all our question in this topic and helping us out in providing some useful information from the time the assignment was given until it has been submitted.

Furthermore, we would like to say thank you to our beloved family who supported us since we started do this assignment. Lastly, we want to thank everyone who were responsible for the making of each online source, and those related to the libraries and bookstores where we conducted our report and for all of them that had helped us in completing the assignment.

Table of Contents

1.0 Introduction

Identification of the Real-World Form

This project is developed as part of the IMS566: Advanced Web Design Development and Content Management course.

The primary purpose of this project is to apply advanced knowledge of web design, content management, and system development to create a fully functional web application with database interaction. For this assignment, the goal is to transform the manual flight booking and scheduling process into a digitalized Flight Booking System.

System Overview

The **FlyHigh Flight Booking System** is an advanced, full-stack web application developed to modernize the airline ticketing experience. It addresses the complexity of modern travel by offering a streamlined, user-friendly interface for passengers and a robust management console for administrators. The system is architected using CakePHP 5.2, a rapid development PHP framework that ensures security, maintainability, and scalability.

Purpose & Problem Statement

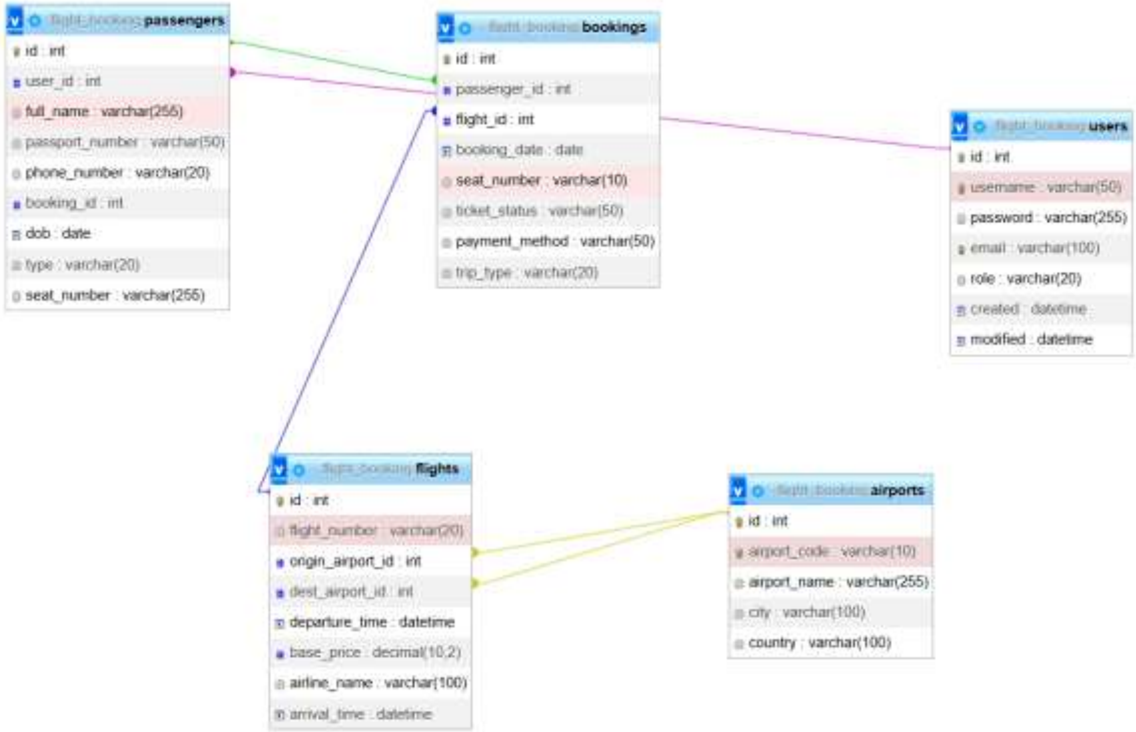
Traditional booking systems are often clunky and difficult to navigate on mobile devices. FlyHigh solves this by implementing a **Mobile-First** design strategy.

- **For Passengers/User** : It simplifies the complex logic of multi-leg journeys (Round Trips), dynamic pricing based on cabin class, and instant PDF itinerary generation.
- **For Administrators**: It centralizes the management of flight schedules, airport codes, and passenger manifests, reducing manual data entry errors.

2.0 GitHub Repository Link

https://github.com/megatnaufal/flight_booking.git

3.0 Entity-Relationship Diagram (ERD)



Entity Descriptions

Entity	Purpose
USERS	Stores registered accounts (customers & admins). The role field distinguishes between user and admin.
FLIGHTS	Contains flight schedules with origin/destination airports, times, and base pricing.
BOOKINGS	The central transaction table linking users to flights. Tracks booking status and total cost.
PASSENGERS	Stores traveler details (name, passport) for each booking. One booking can have multiple passengers.
AIRPORTS	Reference data for Malaysian airports (KLIA, Penang, Langkawi, etc.).

4.0 System Requirements (software, version compatibility, dependencies)

To ensure optimal performance and security, the system relies on a specific stack of technologies.

Software Environment:

- **Operating System**: Windows 10/11 (Development), capable of Linux/Unix deployment.
- **Web Server**: Compatible with **Apache** (via .htaccess mod_rewrite support) or **Nginx**.
- **Database**: **MySQL 5.7+** or **MariaDB 10+** is required for its relational data integrity (Foreign Keys for bookings/passengers).
- **PHP Runtime**: **PHP 8.1+** is strictly required to support CakePHP 5 features like named arguments and attributes.
- **Dependency Manager:** Composer 2.x

Core Dependencies & Libraries

Defined in { } composer.json, these libraries power critical features:

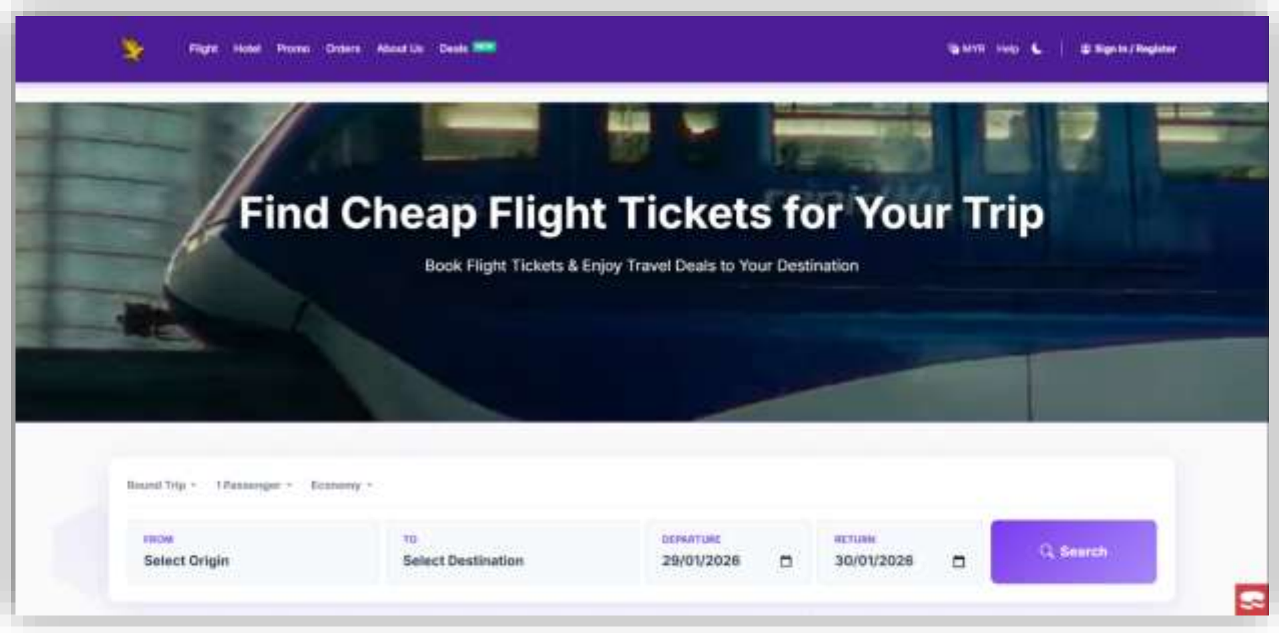
1. **cakephp/cakephp (^5.2)**: The core framework providing the MVC structure, ORM (Object-Relational Mapping), and Security utilities.
2. **cakephp/migrations (^4.0)**: Used for version-controlled database changes, ensuring the database schema stays in sync with the code across different environments.
3. **mobiledetect/mobiledetectlib (^4.8)**: Used in AppController to detect if a user is on a mobile device, allowing the View layer to conditionally render optimized layouts.

5.0 User Interface Overview (layout, navigation)

The UI allows users to intuitively navigate the complex booking flow.

Design System

- **Framework**: **Bootstrap 5.3** provides the responsive grid system.
- **Typography**: The **“Inter”** font family (Google Fonts) was chosen for its high legibility on digital screens.
- **Theme Engine**: A robust **Dark Mode** is implemented in **“templates/layout/default.php."** It uses a **“.dark-mode”** class on the **“<body>”** tag to override global CSS variables, changing backgrounds to #0f0f23 (Deep Navy) and text to #e2e8f0 (Soft White).
- **Visual Flair**: A custom **Hexagon Animation** (.hexagon-bg) uses CSS **@keyframes** to create floating geometric shapes in the background, giving the site a modern, tech-forward feel.



Find Cheap Flight Tickets for Your Trip

Book Flight Tickets & Enjoy Travel Deals to Your Destination

Select Origin

Select Destination

29/01/2026

30/01/2026

Navigation Hierarchy

- **Top Header:** Contains the brand logo, main navigation links (Flight, Hotel, Promo), currency selector, a Dark Mode Toggle, and User Account/Login actions.

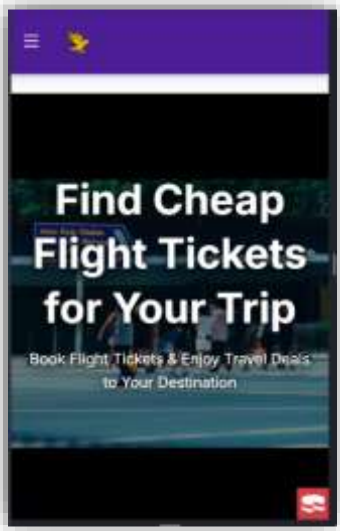
(**Notes** : The "**Promo**," "**Orders**," and "**Deals**" features are included as UI placeholders to demonstrate the intended system layout and navigation)

- **Hero Section**: Dynamic background with a gradient overlay (#4C1D95 to #7C3AED) featuring a search widget.
- **Content Area:** Card-based layouts for flight results and dashboards to segregate information clearly.

Footer: Simple navigation and copyright information.

(**Notes**: All Items in the footer section serve as aesthetic placeholders to represent the intended system architecture.)

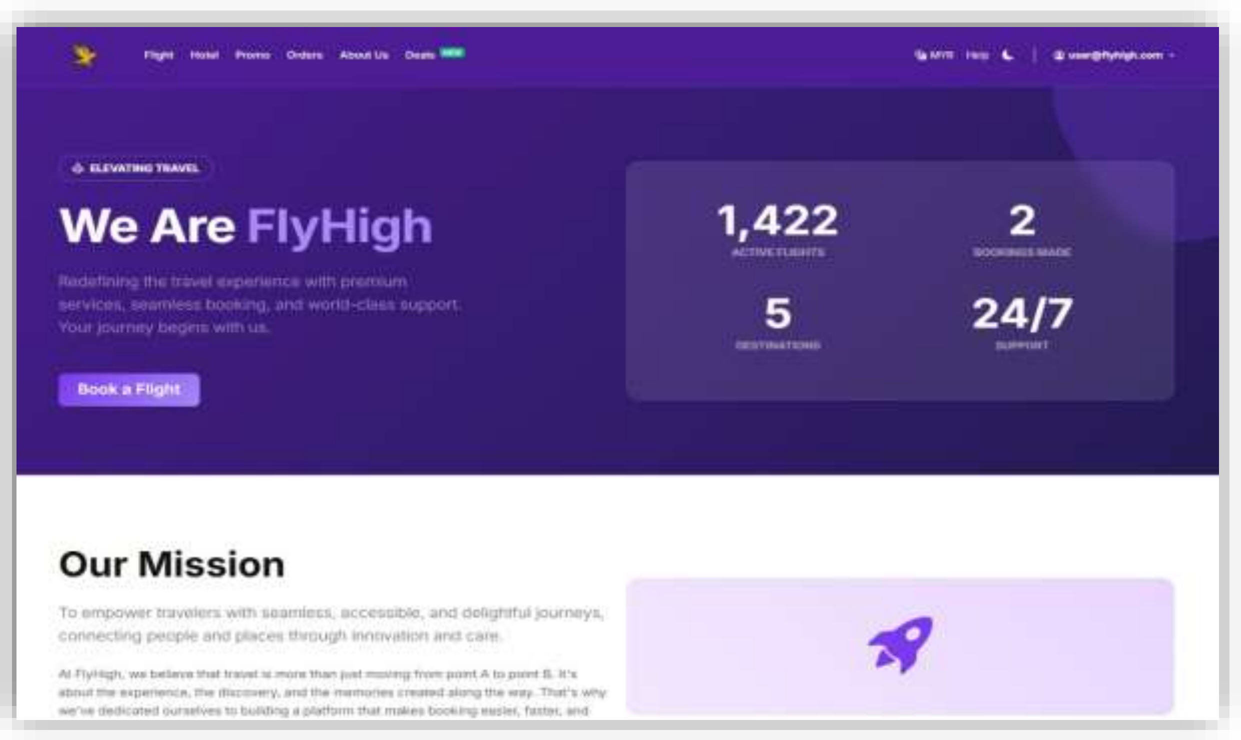
Off-Canvas Mobile Menu: Hidden by default on desktop, this slides in from the left on mobile devices (Burger Menu) , providing full access to the menu without cluttering the small screen.



Find Cheap Flight Tickets for Your Trip

Book Flight Tickets & Enjoy Travel Deals to Your Destination

About Us Page



We Are FlyHigh

Redefining the travel experience with premium services, seamless booking, and world-class support. Your journey begins with us.

1,422 ACTIVE FLIGHTS

2 BOOKINGS MADE

5 DESTINATIONS

24/7 SUPPORT

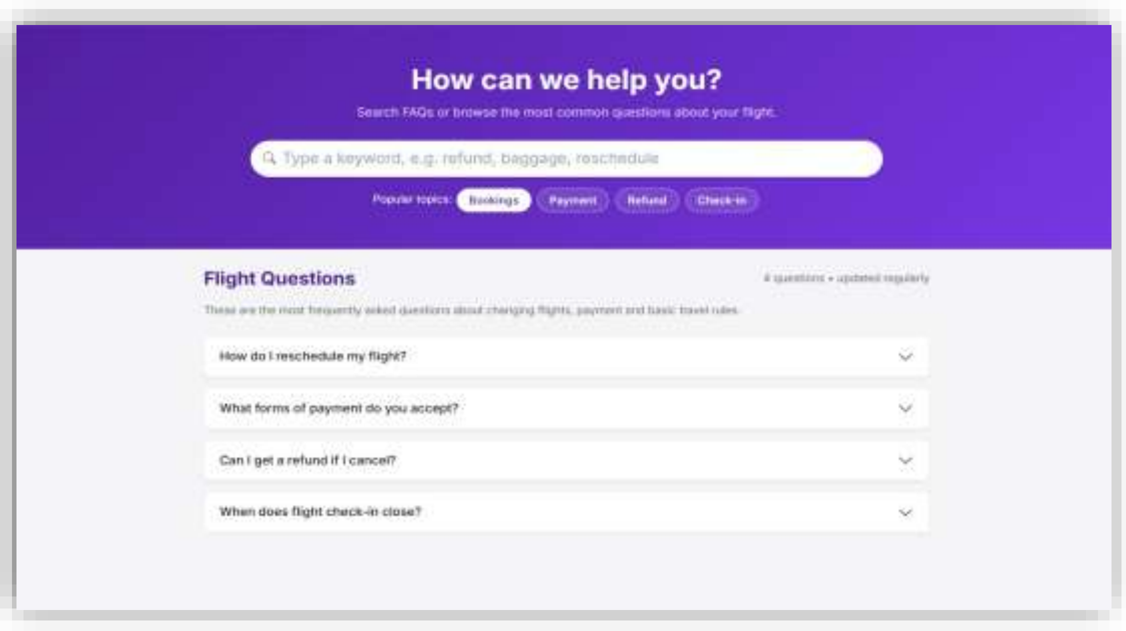
Our Mission

To empower travelers with seamless, accessible, and delightful journeys, connecting people and places through innovation and care.

At FlyHigh, we believe that travel is more than just moving from point A to point B. It's about the experience, the discovery, and the memories created along the way. That's why we've dedicated ourselves to building a platform that makes booking easier, faster, and

- Introduces FlyHigh as a modern, premium flight booking platform with seamless booking and strong customer support.
- Highlights the brand with a bold “We Are FlyHigh” hero banner, a clear “Book a Flight” button, key statistics, and an “Our Mission” section to build user trust.

Help Page



How can we help you?

Search FAQs or browse the most common questions about your flight.

Flight Questions

4 questions • updated regularly

These are the most frequently asked questions about changing flights, payment and basic travel rules.

How do I reschedule my flight?

What forms of payment do you accept?

Can I get a refund if I cancel?

When does flight check-in close?

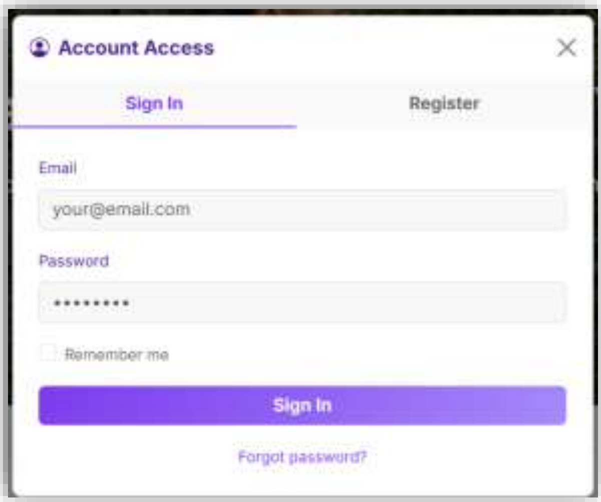
- Provides a dedicated space where users can access guidance, FAQs, and support for booking and managing their flights.
- Helps passengers quickly resolve common issues without direct staff assistance, delivering practical, easy-to-use help that fulfills the promise of seamless booking and strong support.

6.0 Features & Functionalities (with screenshots)

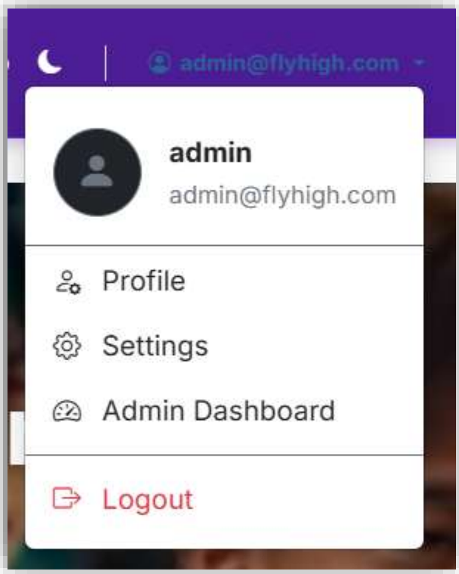
1. User Authentication & Role Management

The UsersController handles secure access.

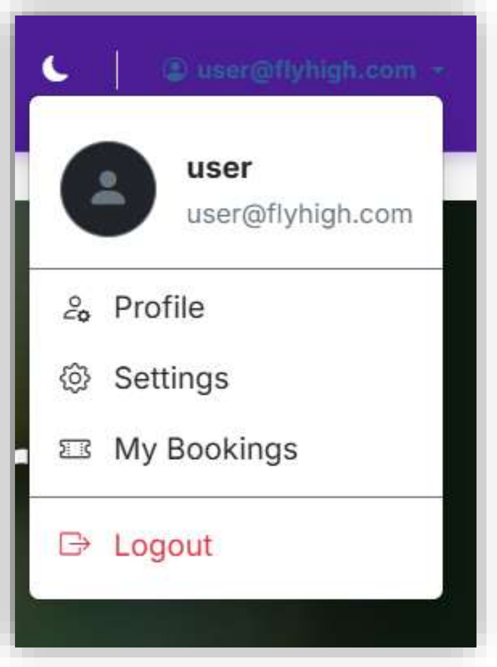
- **Registration**: New users are assigned the user role by default in the add() method.



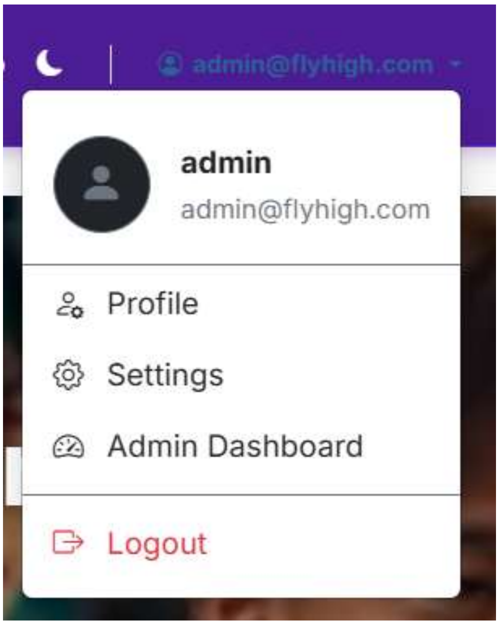
- **Users**: Access to Home, Search, About Us, and Login/Register.



- **Registered Users**: Access to My Bookings, Profile Settings, and Ticket Download.

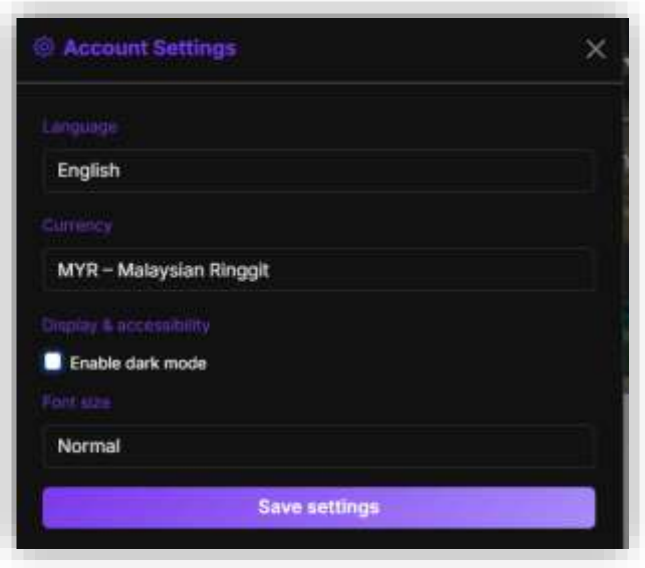


- **Admin Access**: Special routes check if Auth.role === 'admin'. Admin users are redirected to a specialized Dashboard upon login (DashboardsController), while regular users go to the Home page.



Admins: Access to the specialized Dashboard for managing system data.

- **Settings**: Users can update their preferences (Language, Currency) which are stored in their session configuration.



Language

English

Currency

MYR – Malaysian Ringgit

Display & accessibility

□ Enable dark mode

Font size

Normal

2. Advanced Flight Search Engine

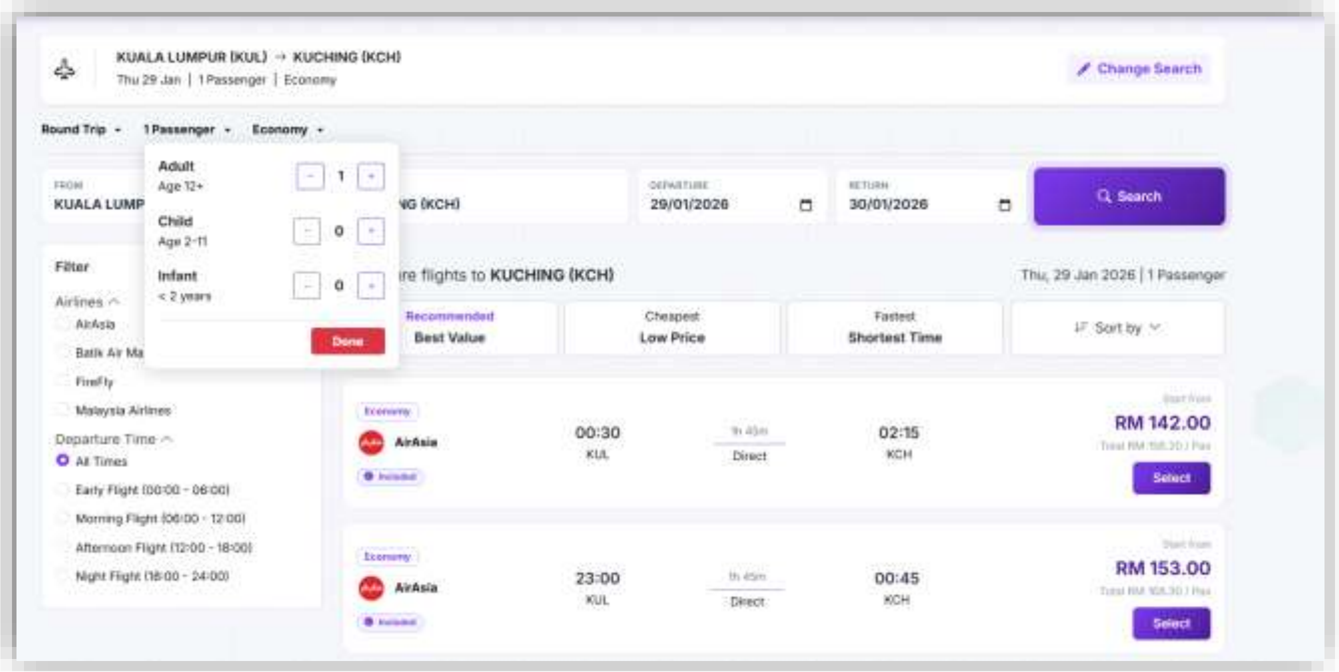
Located in FlightsController.php, the search engine is the heart of the application.

- **Logic**: The _searchFlights method builds a dynamic query based on user input (Origin, Destination, Date).

- **Time Filtering**: Users can filter flights by time of day (Early, Morning, Afternoon, Night). The code uses MySQL HOUR() functions to categorize flights (e.g., HOUR(departure_time) < 6 for Early).

- **Price Sorting**: The system supports sorting results by "Cheapest" (Database ORDER BY base_price ASC) or "Fastest" (Calculated via timestamp difference).

- **Cabin Class Pricing**: If a user selects "Business Class", a multiplier of **2.5x** is effectively applied to the base price in the controller logic.



KUALA LUMPUR (KUL) → KUCHING (KCH)
Thu 29 Jan | 1 Passenger | Economy

Round Trip | 1 Passenger | Economy

Adult Age 12+ 1
Child Age 2-11 0
Infant < 2 years 0

FROM: KUALA LUMP… | …NG (KCH) | DEPARTURE: 29/01/2026 | RETURN: 30/01/2026

…re flights to KUCHING (KCH)
Thu, 29 Jan 2026 | 1 Passenger

Economy AirAsia 00:30 KUL — Direct — 02:15 KCH — RM 142.00

Economy AirAsia 23:00 KUL — Direct — 00:45 KCH — RM 153.00

Notes: To optimize system performance and prevent excessive storage consumption within the database, flight schedules are generated for a strictly limited 5-day range, and users may only select travel dates within this active timeframe. If you need to refresh the flight schedule, seed a new range for future dates, or if the data is not generated automatically, run the command:

```
php bin/cake.php migrations seed --seed FlightSchedulesSeed
```

This command updates the database with new flight data and ensures the schedule remains up to date.

3. Intelligent Booking Workflow

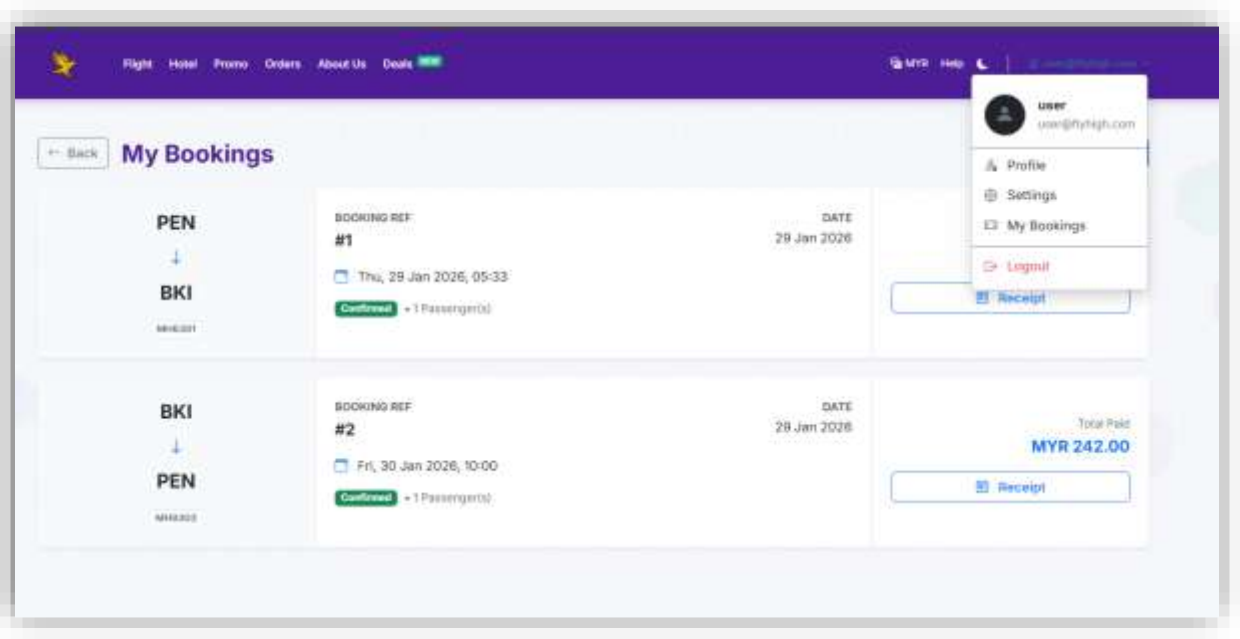
The system manages state across multiple pages using **User Sessions**.

i. **Session Storage**: When a user selects a departure flight, the data is not immediately saved to the database. Instead, FlightsController::selectDeparture writes the data to Session['Flight.Departure']. This allows the user to browse Return flights without losing their initial selection.

ii. **Data Validation**: The BookingsController::add method checks for this session data before allowing entry. If the session expires, the user is safely redirected to the home page.

4. Booking Management & PDF Receipts

- **My Bookings**: A centralized dashboard (BookingsController::myBookings) uses CakePHP's contain method to fetch related data: Bookings -> Flights -> Airports.



My Bookings

PEN ↓ BKI

BOOKING REF
#1

DATE
29 Jan 2026

Thu, 29 Jan 2026, 05:33

Confirmed • 1 Passenger(s)

BKI ↓ PEN

BOOKING REF
#2

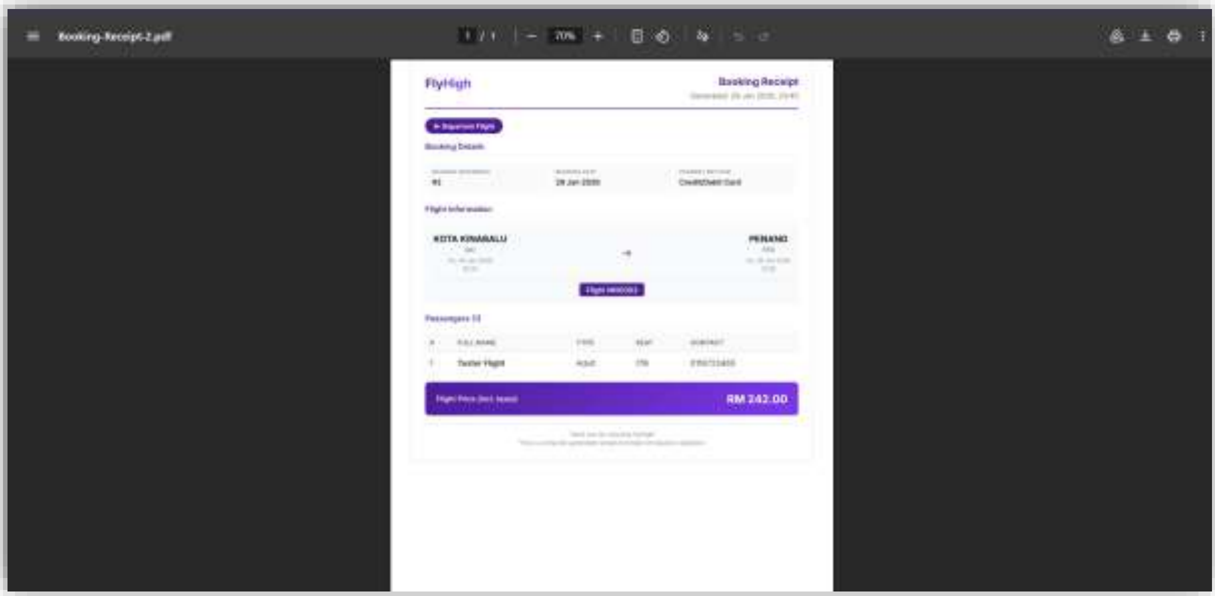
DATE
28 Jan 2026

Fri, 30 Jan 2026, 10:00

Confirmed • 1 Passenger(s)

Total Paid
MYR 242.00

- **PDF Generation**: The downloadReceipt($id) method fetches the full booking hierarchy and renders a specialized "Ajax" layout that converts into a professional PDF document.



Booking-Receipt-2.pdf

FlyHigh

Booking Receipt

Booking Details

#1	[illegible]	Credit/Debit Card

Flight Information

KOTA KINABALU → PENANG

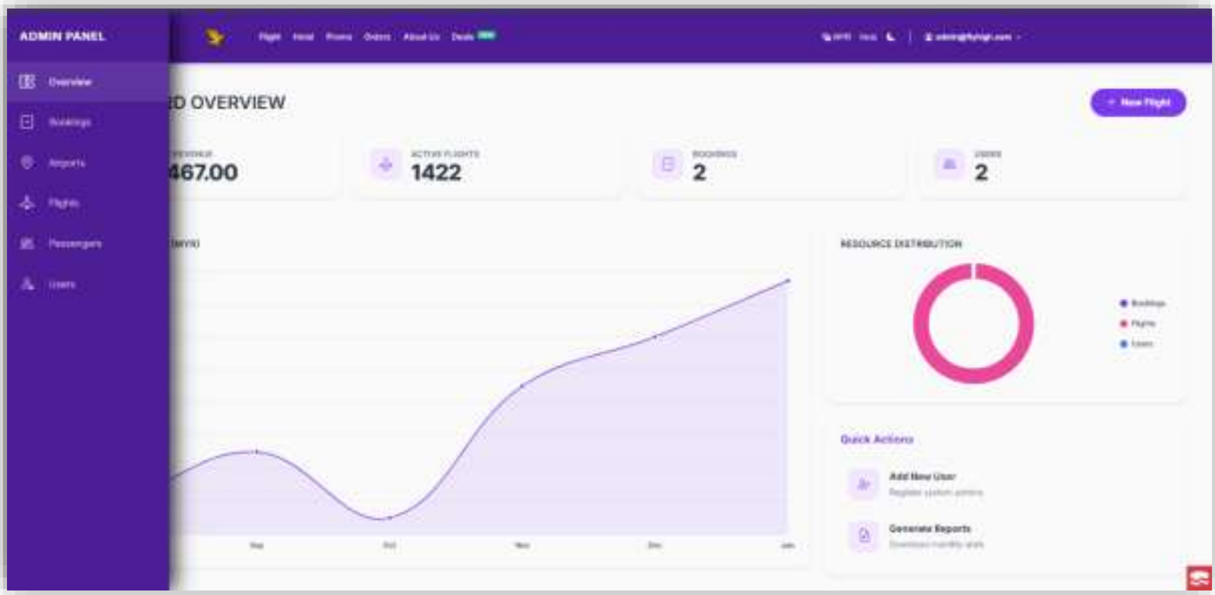
Passengers (1)

#	FULL NAME	TYPE	SEAT	CONTACT
1	Tester Flight	Adult	[illegible]	[illegible]

Flight Price (incl. taxes) RM 242.00

5. Admin Dashboard

- **Chart.js Integration:** Renders revenue trends as interactive line/pie charts.
- **CRUD Operations:** Quick links to manage Flights, Airports, Bookings, Passengers, and Users.
- **Collapsible Sidebar:** Navigation auto-expands on hover for a clean UI.



System Interface: Admin Dashboard Overview

Notes: The **FlyHigh Admin Panel** serves as the central management hub for the flight reservation system. It is designed with a modern, responsive user interface (UI) to provide administrators with efficient control over flight operations, bookings, and user data. The layout consists of a persistent **Sidebar Navigation** for module access and a dynamic **Main Content Area** for data visualization and management.

➢ **Dashboard Overview Module**

The "Overview" page is the landing screen for administrators, designed to provide high-level insights into system performance briefly.

- **Key Performance Indicators (KPI) Cards:** Four prominent cards display real-time statistics to help admins monitor system health:
  - **Monthly Revenue:** Tracks financial earnings (MYR) for the current period.
  - **Active Flights:** Shows the total number of scheduled flights currently active in the database.
  - **Bookings:** Displays the total count of passenger reservations.
  - **Users:** Indicates the number of registered system administrators and staff accounts.
- **Revenue Trend Analytics:** A line chart visualizes financial performance over a 6-month timeline. This graphical representation allows administrators to identify booking trends, seasonal peaks, and revenue dips effectively.
- **Resource Distribution:** A doughnut chart provides a visual breakdown of system data, illustrating the ratio between Bookings, Flights, and Users.
- **Quick Actions Panel:** A dedicated sidebar widget offers shortcuts to frequently used administrative tasks, such as **"Add New User"** and **"Generate Reports" (Download PDF)** streamlining the workflow.

➢ **Data Management Modules**

The system includes specialized sections for managing core entities. Each section follows a consistent design pattern featuring a data table, status indicators, and action controls.

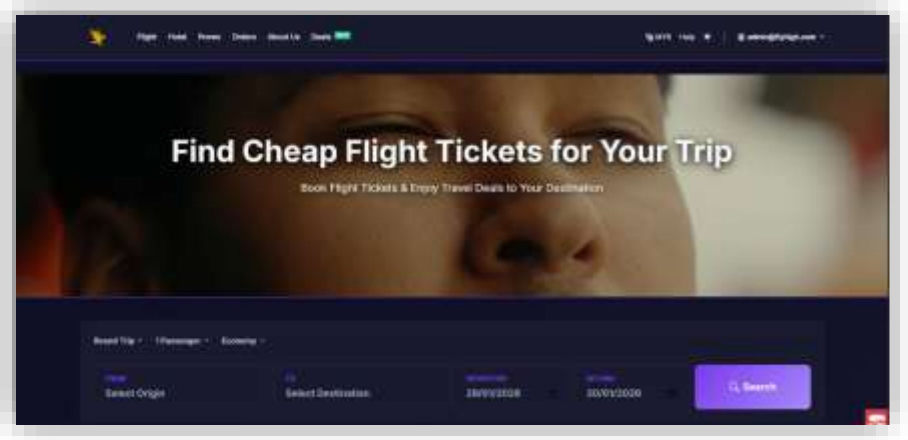
- **Flight Schedules Management:**
  - Lists all available flights with essential details: **Flight No.**, **Origin**, **Destination**, **Departure Time**, and **Base Price**.
  - Admins can use the **"+ New Flight"** button to schedule new routes.
  - Each row provides **View**, **Edit**, and **Delete** actions for granular control over flight records.
- **Bookings Management:**
  - Displays a log of passenger reservations.
  - Key columns include **Passenger Name**, **Flight Code**, **Date**, **Trip Type** (One Way/Round Trip), and **Payment Status**.
  - **Status Badges:** Color-coded badges (e.g., green for "PAID") allow admins to quickly assess payment status.
- **Passenger Database:**
  - Maintains a registry of traveler information including **Full Name**, **Passport Number**, and **Phone Number**.
  - Links passengers to their registered **User Accounts** (or marks them as Guests), facilitating customer relationship management (CRM).
- **System Users & Access Control:**
  - Manages administrative and staff accounts.
  - The table displays **Username**, **Email**, and **Role** (e.g., Admin, User).
  - **Role Badges:** Distinct badges differentiate between "ADMIN" and standard "USER" privileges, ensuring security and proper access control.
- **Airport Terminals:**
  - A reference database for airport codes (e.g., KUL, PEN), names, and locations, ensuring accuracy in flight scheduling.

- ➢ **User Experience (UX) & Design**

- **Navigation:** A collapsible, dark-themed sidebar provides persistent access to all modules, ensuring users can navigate between sections (Overview, Bookings, Flights, etc.) without losing context.
- **Visual Hierarchy:** The use of "card-based" layouts separates different data sets, making the information easy to scan. Critical actions (like "New Flight") are highlighted with distinct colors (e.g., Purple/Red buttons).

6. **Dark Mode :** A system-wide theme toggle implemented with CSS classes and JavaScript

- **CSS Implementation**: Uses body.dark-mode selector to override colors
- **JavaScript Toggle**: The header button toggles the class on click


Find Cheap Flight Tickets for Your Trip
Book Flight Tickets & Enjoy Travel Deals to Your Destination
Round Trip
1 Passenger
Economy
Select Origin
Select Destination
Search

Covers all pages: Home, Search Results, Checkout, Modals, and Form controls.

7.0 Workflow of Form (illustration of the CRUD cycle)

The Booking process demonstrates a complete lifecycle of data manipulation.

1. READ (The Search Phase)

The cycle begins in FlightsController::index and search. The system reads available flight records from the flights table.

- *Code Insight*: $this->Flights->find()->contain(['OriginAirports', 'DestAirports']) ensures that airport names (e.g., "KUL") are displayed instead of just IDs.

2. CREATE (The Booking Phase)

Data persistence happens in BookingsController::add.

- **Transaction Logic**: This is a complex multi-entity save operation.

    1. The **Flight** details are duplicated from the master schedule to specific flights record for this booking (locking the price).
    2. The **Lead Passenger** is saved to the passenger's table.
    3. The **Booking** is created, linking the Passenger ID and Flight ID.
    4. **Additional Passengers** are saved in a loop, linked to the Booking ID.
    5. **Seat Assignment**: The _nextSeat() private method automatically assigns sequential seats (1A, 1B, 1C...) to the party.

3. UPDATE (The Payment Phase)

The record is modified in BookingsController::complete.

- **Status Update**: The system retrieves the pending booking ($this->Bookings->get($id)) and updates the ticket_status from "Pending" to "Confirmed".
- **Payment Method**: The user's selected card payment method or bank (e.g., "Maybank2u") that is appended to the payment record for audit trails.
- **Return Flight Sync**: If it's a Round Trip, the system automatically finds the associated Return Booking ID from the session and updates its status simultaneously.

4. DELETE (Cancellation Phase)

Managed via BookingsController::delete.

- This method accepts only **POST** or **DELETE** requests for security (preventing accidental deletion via URL). Validates that the booking exists before removing it from the database.

8.0 Team Roles & Contributions

This project was collaboratively developed by a dedicated team, with each member focusing on specific core modules to ensure a robust and well-integrated system.

1. Project Manager & Backend Lead (Amsyar & Najmi)

- **Responsibilities**: Oversaw the entire system architecture, database design (Schema migrations), and core business logic implementation.
- **Key Contributions**:
    - Designed the Relational Database schema (Users, Flights, Bookings).
    - Implemented the BookingsController logic for complex multi-leg journeys.
    - Managed the Git repository and merge requests.

2. Frontend Developer & UI/UX Designer (Megat & Faris)

- **Responsibilities**: Focused on the user interface, responsiveness, and overall user experience.
- **Key Contributions**:
    - Developed the "Mobile-First" responsive layout using Bootstrap 5.
    - Created the custom "Dark Mode" theme and Hexagon background animations.
    - Designed the intuitive Booking Progress bar and interactive Flight Search forms.

3. Quality Assurance & Documentation Specialist (All)

- **Responsibilities**: Ensured the system was bug-free and documented all features for the final report.
- **Key Contributions**:
    - Performed rigorous testing of the Booking CRUD cycle.
    - Wrote the User Manual and System Report.
    - Implemented the PDF Receipt generation feature using DomPDF to ensure accurate output.

9.0 Contact Information (Support)

We are committed to providing 24/7 support for all FlyHigh users.

Online Help Center

- **FAQ**: Accessible via the "Help" page, covering common questions like Refunds, Check-in times, and Payment methods.

Direct Support Channels

- **Email**: support@flyhigh.com (24/7 Response)
- **Hotline**: +60 3-8888 1234 (Available 9 AM - 6 PM, Mon-Fri)
- **Office Address**: FlyHigh HQ, Level 25, Menara Tech, Jalan Tun Razak, 50400 Kuala Lumpur, Malaysia.

Staff Contact Information:

Mr. Megat Naufal: +60 11-6404 6283

Mr. Muhammad Najmi: +60 11-6211 7734

Mr. Faris Afizuan: +60 19-950 4432

Mr. Amir Amsyar: +60 14-699 3203

10.0 Conclusion & Reflection

The development of the FlyHigh system highlights the importance of **Session Management** in multi-step e-commerce flows. Managing state between the "Select Departure", "Select Return", and "Passenger Details" pages was a key challenge solved by CakePHP's Session object. The resulting application is not just a CRUD interface, but a complex state machine that guides users securely from search to confirmed itinerary. The clean separation of concerns (MVC) allows for future features such as real-time API integrations with actual airlines to be added with minimal refactoring.

A primary highlight of this project was the implementation of a robust Session Management system within a multi-step e-commerce flow. Managing the persistent state of data between the "Select Departure," "Select Return," and "Passenger Details" pages was a significant technical challenge. By utilizing CakePHP’s Session object, we ensured that user selections remain secure and consistent throughout the booking journey.

Overall, this project demonstrates our team's ability to balance professional UI/UX design with complex backend logic to create a responsive and reliable digital solution.